

FILEID**CONIO

H 4

CCCCCCCC 000000 NN NN IIIIIII 000000
CCCCCCCC 000000 NN NN IIIIIII 000000
CC 00 00 NN NN IIIIIII 00 00
CC 00 00 NNNN NN IIIIIII 00 00
CC 00 00 NNNN NN IIIIIII 00 00
CC 00 00 NN NN IIIIIII 00 00
CC 00 00 NN NN IIIIIII 00 00
CC 00 00 NN NNNN IIIIIII 00 00
CC 00 00 NN NN IIIIIII 00 00
CCCCCCCC 000000 NN NN IIIIIII 000000
CCCCCCCC 000000 NN NN IIIIIII 000000

....
....
....

LL IIIIIII SSSSSSSS
LL IIIIIII SSSSSSSS
LL IIIIIII SSSSSS
LL IIIIIII SSSSSS
LL IIIIIII SS
LL IIIIIII SS
LL IIIIIII SS
LL IIIIIII SS
LLLLLLLLLL IIIIIII SSSSSSSS
LLLLLLLLLL IIIIIII SSSSSSSS

(1) 62 boo\$readprompt - prompt and read input string

```
0000 1 .title CONIO - console input output routines
0000 2 .ident /V04-000/
0000 3 .
0000 4 .
0000 5 ****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 ****
0000 27
0000 28 Facility: system bootstrapping
0000 29
0000 30 Abstract: CONIO provides basic console read, readprompt and write facilities.
0000 31
0000 32 Author: Richard I. Hustvedt, creation date: 27-apr-1978
0000 33
0000 34 Modified by:
0000 35
0000 36 V03-002 WHM0001 Bill Matthews 9-Jul-84
0000 37
0000 38 Add generalized alternate console terminal support
0000 39
0000 40 V03-001 DNC0001 David N. Cutler 29-Dec-83
0000 41
0000 42 Add support for QVSS as the console terminal on MicroVax I.
0000 43
0000 44 Include files:
0000 45
0000 46
0000 47 Sprdef ; define processor registers
0000 48 $ssdef ; define status code values
0000 49
0000 50
0000 51 Equated symbols:
0000 52
0000 53
0000 54 cr = 13 ; character code for carriage return
0000 55 lf = 10 ; character code for line feed
0000 56 control_u = 21 ; character code for control-u
0000 57 control_s = 19 ; control s (xoff)
```

00000011	0000	58	control_a = 17	: control q (xon)
0000007F	0000	59	rubout = 127	: character code for rubout
00000000	0000	60	v_rub = 0	: rubout sequence in progress

```

0000 62 .sbttl boo$readprompt - prompt and read input string
0000 63 :+
0000 64 :+
0000 65 :+
0000 66 :+
0000 67 :+
0000 68 :+
0000 69 :+
0000 70 :+
0000 71 :+
0000 72 :+
0000 73 :+
0000 74 :+
0000 75 :+
0000 76 :+
0000 77 :+
0000 78 :+
0000 79 :+
0000 80 :+
0000 81 :+
0000 82 :+
0000 83 :+
0000 84 :+
0000 85 :+
0000 86 :+
0000 87 :+
0000 88 :+
0000 89 :+
0000 90 :+
0000 91 :+
0000 92 :+
0000 93 :+
0000 94 :+
0000 95 :+
0000 96 :+
0000 97 :+
0000 98 .psect $conio,byte
0000 99 .entry boo$readprompt,^m<r2,r4,r8,r9>
0314 0000 100 10$: movl prompt(ap),r8 ;get prompt string address
      D0 0002 101      clrl r4 ;clear control flags
      54 0006 102 20$: movzbl (r8)+,r0 ;get next output character
      88 9A 0008 103      beql 30$ ;if eql none
      05 13 000B 104      bsbw con$putchar ;output character
      F6 11 0010 105      brb   20$ ;:
      0000 0000 106      movzbl size(ap),r2 ;maximum number of characters to read
      0000 0000 107 30$: beql 120$ ;if eql none
      0000 0000 108      movl buf(ap),r9 ;set address of input buffer
      0000 0000 109      clrb (r9)+ ;initialize string count
      0075 30 000D 104      subtr r2,40$ ;decrement and test character count
      0000 0000 110      brb   110$ ;end of read
      0000 0000 111      movl r2,40$ ;get a character
      0000 0000 112      bicb3 #^x80,r0,r8 ;clear parity bit
      0000 0000 113      cmpb #rubout,r8 ;rubout?
      0000 0000 114 40$: bneq 80$ ;if neq no
      0000 0000 115      movzbl -(r9),r8 ;get character to rubout
      0000 0000 116      0000 0000 117      0000 0000 118
      0000 0000 119      0000 0000 120      0000 0000 121
      0000 0000 122      0000 0000 123      0000 0000 124
      0000 0000 125      0000 0000 126      0000 0000 127
      0000 0000 128      0000 0000 129      0000 0000 130
      0000 0000 131      0000 0000 132      0000 0000 133
      0000 0000 134      0000 0000 135      0000 0000 136
      0000 0000 137      0000 0000 138      0000 0000 139
      0000 0000 140      0000 0000 141      0000 0000 142
      0000 0000 143      0000 0000 144      0000 0000 145
      0000 0000 146      0000 0000 147      0000 0000 148
      0000 0000 149      0000 0000 150      0000 0000 151
      0000 0000 152      0000 0000 153      0000 0000 154
      0000 0000 155      0000 0000 156      0000 0000 157
      0000 0000 158      0000 0000 159      0000 0000 160
      0000 0000 161      0000 0000 162      0000 0000 163
      0000 0000 164      0000 0000 165      0000 0000 166
      0000 0000 167      0000 0000 168      0000 0000 169
      0000 0000 170      0000 0000 171      0000 0000 172
      0000 0000 173      0000 0000 174      0000 0000 175
      0000 0000 176      0000 0000 177      0000 0000 178
      0000 0000 179      0000 0000 180      0000 0000 181
      0000 0000 182      0000 0000 183      0000 0000 184
      0000 0000 185      0000 0000 186      0000 0000 187
      0000 0000 188      0000 0000 189      0000 0000 190
      0000 0000 191      0000 0000 192      0000 0000 193
      0000 0000 194      0000 0000 195      0000 0000 196
      0000 0000 197      0000 0000 198      0000 0000 199
      0000 0000 200      0000 0000 201      0000 0000 202
      0000 0000 203      0000 0000 204      0000 0000 205
      0000 0000 206      0000 0000 207      0000 0000 208
      0000 0000 209      0000 0000 210      0000 0000 211
      0000 0000 212      0000 0000 213      0000 0000 214
      0000 0000 215      0000 0000 216      0000 0000 217
      0000 0000 218      0000 0000 219      0000 0000 220
      0000 0000 221      0000 0000 222      0000 0000 223
      0000 0000 224      0000 0000 225      0000 0000 226
      0000 0000 227      0000 0000 228      0000 0000 229
      0000 0000 230      0000 0000 231      0000 0000 232
      0000 0000 233      0000 0000 234      0000 0000 235
      0000 0000 236      0000 0000 237      0000 0000 238
      0000 0000 239      0000 0000 240      0000 0000 241
      0000 0000 242      0000 0000 243      0000 0000 244
      0000 0000 245      0000 0000 246      0000 0000 247
      0000 0000 248      0000 0000 249      0000 0000 250
      0000 0000 251      0000 0000 252      0000 0000 253
      0000 0000 254      0000 0000 255      0000 0000 256
      0000 0000 257      0000 0000 258      0000 0000 259
      0000 0000 260      0000 0000 261      0000 0000 262
      0000 0000 263      0000 0000 264      0000 0000 265
      0000 0000 266      0000 0000 267      0000 0000 268
      0000 0000 269      0000 0000 270      0000 0000 271
      0000 0000 272      0000 0000 273      0000 0000 274
      0000 0000 275      0000 0000 276      0000 0000 277
      0000 0000 278      0000 0000 279      0000 0000 280
      0000 0000 281      0000 0000 282      0000 0000 283
      0000 0000 284      0000 0000 285      0000 0000 286
      0000 0000 287      0000 0000 288      0000 0000 289
      0000 0000 290      0000 0000 291      0000 0000 292
      0000 0000 293      0000 0000 294      0000 0000 295
      0000 0000 296      0000 0000 297      0000 0000 298
      0000 0000 299      0000 0000 300      0000 0000 301
      0000 0000 302      0000 0000 303      0000 0000 304
      0000 0000 305      0000 0000 306      0000 0000 307
      0000 0000 308      0000 0000 309      0000 0000 310
      0000 0000 311      0000 0000 312      0000 0000 313
      0000 0000 314      0000 0000 315      0000 0000 316
      0000 0000 317      0000 0000 318      0000 0000 319
      0000 0000 320      0000 0000 321      0000 0000 322
      0000 0000 323      0000 0000 324      0000 0000 325
      0000 0000 326      0000 0000 327      0000 0000 328
      0000 0000 329      0000 0000 330      0000 0000 331
      0000 0000 332      0000 0000 333      0000 0000 334
      0000 0000 335      0000 0000 336      0000 0000 337
      0000 0000 338      0000 0000 339      0000 0000 340
      0000 0000 341      0000 0000 342      0000 0000 343
      0000 0000 344      0000 0000 345      0000 0000 346
      0000 0000 347      0000 0000 348      0000 0000 349
      0000 0000 350      0000 0000 351      0000 0000 352
      0000 0000 353      0000 0000 354      0000 0000 355
      0000 0000 356      0000 0000 357      0000 0000 358
      0000 0000 359      0000 0000 360      0000 0000 361
      0000 0000 362      0000 0000 363      0000 0000 364
      0000 0000 365      0000 0000 366      0000 0000 367
      0000 0000 368      0000 0000 369      0000 0000 370
      0000 0000 371      0000 0000 372      0000 0000 373
      0000 0000 374      0000 0000 375      0000 0000 376
      0000 0000 377      0000 0000 378      0000 0000 379
      0000 0000 380      0000 0000 381      0000 0000 382
      0000 0000 383      0000 0000 384      0000 0000 385
      0000 0000 386      0000 0000 387      0000 0000 388
      0000 0000 389      0000 0000 390      0000 0000 391
      0000 0000 392      0000 0000 393      0000 0000 394
      0000 0000 395      0000 0000 396      0000 0000 397
      0000 0000 398      0000 0000 399      0000 0000 400
      0000 0000 401      0000 0000 402      0000 0000 403
      0000 0000 404      0000 0000 405      0000 0000 406
      0000 0000 407      0000 0000 408      0000 0000 409
      0000 0000 410      0000 0000 411      0000 0000 412
      0000 0000 413      0000 0000 414      0000 0000 415
      0000 0000 416      0000 0000 417      0000 0000 418
      0000 0000 419      0000 0000 420      0000 0000 421
      0000 0000 422      0000 0000 423      0000 0000 424
      0000 0000 425      0000 0000 426      0000 0000 427
      0000 0000 428      0000 0000 429      0000 0000 430
      0000 0000 431      0000 0000 432      0000 0000 433
      0000 0000 434      0000 0000 435      0000 0000 436
      0000 0000 437      0000 0000 438      0000 0000 439
      0000 0000 440      0000 0000 441      0000 0000 442
      0000 0000 443      0000 0000 444      0000 0000 445
      0000 0000 446      0000 0000 447      0000 0000 448
      0000 0000 449      0000 0000 450      0000 0000 451
      0000 0000 452      0000 0000 453      0000 0000 454
      0000 0000 455      0000 0000 456      0000 0000 457
      0000 0000 458      0000 0000 459      0000 0000 460
      0000 0000 461      0000 0000 462      0000 0000 463
      0000 0000 464      0000 0000 465      0000 0000 466
      0000 0000 467      0000 0000 468      0000 0000 469
      0000 0000 470      0000 0000 471      0000 0000 472
      0000 0000 473      0000 0000 474      0000 0000 475
      0000 0000 476      0000 0000 477      0000 0000 478
      0000 0000 479      0000 0000 480      0000 0000 481
      0000 0000 482      0000 0000 483      0000 0000 484
      0000 0000 485      0000 0000 486      0000 0000 487
      0000 0000 488      0000 0000 489      0000 0000 490
      0000 0000 491      0000 0000 492      0000 0000 493
      0000 0000 494      0000 0000 495      0000 0000 496
      0000 0000 497      0000 0000 498      0000 0000 499
      0000 0000 500      0000 0000 501      0000 0000 502
      0000 0000 503      0000 0000 504      0000 0000 505
      0000 0000 506      0000 0000 507      0000 0000 508
      0000 0000 509      0000 0000 510      0000 0000 511
      0000 0000 512      0000 0000 513      0000 0000 514
      0000 0000 515      0000 0000 516      0000 0000 517
      0000 0000 518      0000 0000 519      0000 0000 520
      0000 0000 521      0000 0000 522      0000 0000 523
      0000 0000 524      0000 0000 525      0000 0000 526
      0000 0000 527      0000 0000 528      0000 0000 529
      0000 0000 530      0000 0000 531      0000 0000 532
      0000 0000 533      0000 0000 534      0000 0000 535
      0000 0000 536      0000 0000 537      0000 0000 538
      0000 0000 539      0000 0000 540      0000 0000 541
      0000 0000 542      0000 0000 543      0000 0000 544
      0000 0000 545      0000 0000 546      0000 0000 547
      0000 0000 548      0000 
```

02 54	DC 00	13 E2	0034 0036	119 120	beql bbss	30\$ #v_rub,r4,70\$:if eqi none :set start of rubout sequence		
	40	10	003A	121	bsbb	out\$bslsh	:output back slash		
	44	10	003C	122	70\$:	bsbb	:output rubbed out character		
	52	D6	003E	123	incl	r2	:adjust remaining character count		
	E1	11	0040	124	brb	40\$:		
02 54	00 34	E5 10	0042 0046	125 127	bbcc bbbb	#v_rub,r4,90\$ out\$bslsh	:terminate rubout sequence :output backslash		
	58	15	91	0048	128	90\$:	cmpb	:control_u,r8	
03 58	06	B5	13	0048	129		beql	10\$	
	58	20	8A	004D	130		bbc	#6,r8,100\$	
	58	0D	91	0051	131		bicb	#32,r8	
	0C	13	0054	132	100\$:	c:pb	:convert to upper case		
	52	D5	0057	133		beql	#cr,r8		
	C6	13	0059	134		tstl	:carriage return?		
	23	10	005A	135		beql	r2		
	89	58	90	005D	136		beql	40\$	
	BE	52	F4	0062	137		bsbb	:any space left in buffer?	
				0065	138		movb	:if eqi no	
				0065	139		sobgeq	:echo character	
	58	0D	9A	0065	140	110\$:	movzbl	:buffer new character	
	18	10	0068	141		bsbb	ret	:reduce space remaining (always loop)	
	50	0A	9A	006A	142		movzbl	:set carriage return character	
	16	10	006D	143		bsbb			
0C BC	59 0C	AC	C2	006F	144		movzbl	:yes send line feed also	
	59	01	83	0073	145		bsbb	:output character in r0	
	50	01	3C	0078	146	120\$:	subl	:compute character count + 1	
	04	007B	007C	147			subb3	:set actual character count	
				148			movzwl	:return normal completion status	
				007C	149		ret	:	
	50	5C	8F	9A	007C	out\$bslsh:			
	03	11	0080	0082	150		movzbl	:output back slash	
				0082	151		brb	:set character code	
	50	58	9A	0082	152		con\$putchar	;and output it	
				0085	153	outr8:	movzbl	:get character to output	
				0085	154		r8,r0		
			0000	31	0085	155	con\$putchar::	:output character in r0	
				0088	156		brw	:console terminal output vector	
	18	51	20	DB	0088	158	10\$:	mfpr	:receiver ready?
		51	07	E1	0088	159		bbc	:if clr, receiver not ready
		51	21	DB	008F	160		mfpr	:read input character.
13	51	07	00	ED	0092	161		cmpzv	:control-s?
		11	12	0097	162		bneq	:if neq no	
	F9	51	20	DB	0099	163	20\$:	mfpr	:receiver ready?
		51	07	E1	009C	164		bbc	:if clr, receiver not ready
11	51	07	00	ED	00A0	165		mfpr	:read input character
		EF	12	00A3	166		cmpzv	:is it a control-q?	
	F9	51	22	DB	00AA	168	30\$:	bneq	:no, wait for another character.
		51	07	E1	00AD	169		mfpr	:transmitter done?
	23	50	DA	00B1	170		tbc	:if clr, transmitter not done	
		05	00B4	171		mtpr	r0,#pr\$_txdb	:write output character	
			0085	172		rsb		:return	
			0085	173					
	0000	31	0085	174	con\$getchar::	brw	w^10\$:console terminal input vector	

```
F9 50 20 00B8 176
      DB 00B8 177 10$:    mfpr    #pr$_rxcs,r0      ;receiver ready?
F9 50 07 E1 00B8 178          bbc      #7,r0,10$      ;if clr, receiver not ready
      DB 00BF 179          mfpr    #pr$_rxdb,r0      ;read input character
      05 00C2 180          rsb      ;return
      00C3 181
      00C3 182 con$owncty:: ;these routines are noops in SYSBOOT
      00C3 183 con$releasecty:: ;these routines are noops in SYSBOOT
      05 00C3 184          rsb
      00C4 185
      00C4 186          .end
```

B00\$READPROMPT
 BUF
 CON\$GETCHAR
 CON\$OWNCTY
 CON\$PUTCHAR
 CON\$RELEASECTY
 CONTROL_Q
 CONTROL_S
 CONTROL_U
 CR
 LF
 OPS_ACBD
 OPS_ACBF
 OPS_ACBG
 OPS_ACBH
 OPS_ADDD2
 OPS_ADDD3
 OPS_ADDF2
 OPS_ADDF3
 OPS_ADDG2
 OPS_ADDG3
 OPS_ADDH2
 OPS_ADDH3
 OPS_ADDP4
 OPS_ADDP6
 OPS_ASHP
 OPS_CLRD
 OPS_CLRFD
 OPS_CLRGR
 OPS_CLRHR
 OPS_CMPD
 OPS_CMPF
 OPS_CMPG
 OPS_CMPH
 OPS_CMPP3
 OPS_CMPP4
 OPS_CRC
 OPS_CVTBD
 OPS_CVTBF
 OPS_CVTBG
 OPS_CVTBH
 OPS_CVTDB
 OPS_CVTDF
 OPS_CVTDH
 OPS_CVTDL
 OPS_CVTDW
 OPS_CVTFB
 OPS_CVTFD
 OPS_CVTFG
 OPS_CVTFH
 OPS_CVTFL
 OPS_CVTFW
 OPS_CVTGB
 OPS_CVTGF
 OPS_CVTGH
 OPS_CVTGL
 OPS_CVTGW

- console input output routines

= 00000000	RG	02	OPS_CVTHB	= 000068FD
= C000000C			OPS_CVTHD	= 0000F7FD
= 000000B5	RG	02	OPS_CVTHF	= 0000F6FD
= 000000C3	RG	02	OPS_CVTHG	= 000076FD
= 00000085	RG	02	OPS_CVTHL	= 00006AFD
= 000000C3	RG	02	OPS_CVTHW	= 000069FD
= 00000011			OPS_CVTL0	= 0000006E
= 00000013			OPS_CVTLF	= 0000004E
= 00000015			OPS_CVTLG	= 00004EFD
= 0000000D			OPS_CVTLH	= 00006EFD
= 0000000A			OPS_CVTLP	= 000000F9
= 00000006F			OPS_CVTPL	= 00000036
= 00000004F			OPS_CVTPS	= 00000008
= 00004FFD			OPS_CVTPT	= 00000024
= 00006FFD			OPS_CVTRDL	= 0000006B
= 00000060			OPS_CVTRFL	= 0000004B
= 00000061			OPS_CVTRGL	= 000048FD
= 00000040			OPS_CVTRHL	= 00006BFD
= 00000041			OPS_CVTSP	= 00000009
= 000040FD			OPS_CVTTP	= 00000026
= 000041FD			OPS_CVTWD	= 0000006D
= 000060FD			OPS_CVTWF	= 0000004D
= 000061FD			OPS_CVTWG	= 00004DFD
= 00000020			OPS_CVTWH	= 00006DFD
= 00000021			OPS_DIVD2	= 00000066
= 000000F8			OPS_DIVD3	= 00000067
= 0000007C			OPS_DIVF2	= 00000046
= 000000D4			OPS_DIVF3	= 00000047
= 0000007C			OPS_DIVG2	= 000046FD
= 00007CFD			OPS_DIVG3	= 000047FD
= 00000071			OPS_DIVH2	= 000066FD
= 00000051			OPS_DIVH3	= 000067FD
= 000051FD			OPS_DIVP	= 00000027
= 000071FD			OPS_EDITPC	= 00000038
= 00000035			OPS_EMODD	= 00000074
= 00000037			OPS_EMODF	= 00000054
= 0000000B			OPS_EMODG	= 000054FD
= 0000006C			OPS_EMODH	= 000074FD
= 0000004C			OPS_MATCHC	= 00000039
= 00004CFD			OPS_MNEGD	= 00000072
= 00006CFD			OPS_MNEGF	= 00000052
= 00000068			OPS_MNEGG	= 000052FD
= 00000076			OPS_MNEGH	= 000072FD
= 000032FD			OPS_MOVD	= 00000070
= 0000006A			OPS_MOVF	= 00000050
= 00000069			OPS_MOVG	= 000050FD
= 00000048			OPS_MOVH	= 000070FD
= 00000056			OPS_MOVP	= 00000034
= 000099FD			OPS_MOVTC	= 0000002E
= 000098FD			OPS_MOVTUC	= 0000002F
= 0000004A			OPS_MULD2	= 00000064
= 00000049			OPS_MULD3	= 00000065
= 000048FD			OPS_MULF2	= 00000044
= 000033FD			OPS_MULF3	= 00000045
= 000056FD			OPS_MULG2	= 000044FD
= 00004AFD			OPS_MULG3	= 000045FD
= 000049FD			OPS_MULH2	= 000064FD

```

OPS_MULH3      = 000065FD
OPS_MULP      = 00000025
OPS_POLYD      = 00000075
OPS_POLYF      = 00000055
OPS_POLYG      = 000055FD
OPS_POLYH      = 000075FD
OPS_SCANC      = 0000002A
OPS_SKPC      = 0000003B
OPS_SPANC      = 0000002B
OPS_SUBD2      = 00000062
OPS_SUBD3      = 00000063
OPS_SUBF2      = 00000042
OPS_SUBF3      = 00000043
OPS_SUBG2      = 000042FD
OPS_SUBG3      = 000043FD
OPS_SUBH2      = 000062FD
OPS_SUBH3      = 000063FD
OPS_SUBP4      = 00000022
OPS_SUBP6      = 00000023
OPS_TSTD       = 00000073
OPS_TSTF       = 00000053
OPS_TSTG       = 000053FD
OPS_TSTH       = 000073FD
OUTBSLH       = 0000007C R 02
OUTR8         = 00000082 R 02
PRS_RXCS       = 00000020
PRS_RXDB       = 00000021
PRS_TXCS       = 00000022
PRS_TXDB       = 00000023
PROMPT        = 00000004
RUBOUT        = 0000007F
SIZE          = 00000008
SSS_NORMAL     = 00000001
V_RUB         = 00000000

```

! Psect synopsis !

PSECT name

	Allocation	PSECT No.	Attributes								
. ABS .	00000000	(0.) 00 (0.)	NOPIC USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC BYTE
\$ABSS	00000000	(0.) 01 (1.)	NOPIC USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC BYTE
\$CONIO	000000C4	(196.) 02 (2.)	NOPIC USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC BYTE

! Performance indicators !

Phase

	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.07	00:00:00.39
Command processing	133	00:00:00.79	00:00:04.76
Pass 1	458	00:00:12.51	00:00:24.90
Symbol table sort	0	00:00:01.33	00:00:02.51
Pass 2	54	00:00:03.68	00:00:06.87
Symbol table output	18	00:00:00.14	00:00:00.34

CONIO VAX-11 Macro Run Statistics

- console input output routines

D 5

15-SEP-1984 23:48:08 VAX/VMS Macro V04-00
4-SEP-1984 23:03:51 [BOOTS.SRC]CONIO.MAR;1

Page 8
(1)

Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	705	00:00:18.55	00:00:39.80

The working set limit was 1650 pages

59843 bytes (117 pages) of virtual memory were used to buffer the intermediate code. There were 50 pages of symbol table space allocated to hold 865 non-local and 14 local symbols. 2938 source lines were read in Pass 1, producing 16 object records in Pass 2. 136 pages of virtual memory were used to define 135 macros.

Macro library statistics

Macro library name

Macros defined

- \$255\$SDUA28:[BOOTS.OBJ]BOOTS.MLB;1
- \$255\$SDUA28:[SYS.OBJ]LIB.MLB;1
- \$255\$SDUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

918 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:CONIO/OBJ=OBJ\$:CONIO MASDS:[EMULAT.SRC]MISSING/UPDATE=(MASDS:[EMULAT.ENH]MISSING)+MASDS:[BOOTS.SRC]CONIO/UPDATE=(MASD

0038 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

